

- 4 -

REMARKS

This response is to the official action mailed in the above-referenced case on June 18, 2007. Claims 1-11 are standing for examination. Claims 1, and 3-9 are rejected under 35 U.S.C. 103(a) as being unpatentable over Batch as taught by "Not a Batch Language: A Control Language!", E.H. Bristol, published May 1995, in view of the implementation of the methodology of Object Oriented as taught by Object-Oriented Modelling and Simulation of Batch Plants (Wollhaf et al.) from November 1955, hereinafter Wollhaf.

In response to the Examiner's rejections and objections, applicant herein presents extensive arguments clearly showing where the art of Batch and Wollhaf fail to teach all of the limitations of applicant's claims, as amended.

Applicant's claims clearly recite that the above models (abstractions) are instantiated with appropriate input data parameters generating appropriate instances of batch functions and function operations wherein the generated instances are automatically executable as part of a run sequence of the purpose-specific batch program. In applicant's invention, as claimed, a user may simply initiate a model abstraction of the program and then leave it up to the frame work and code generators to automatically implement a correct batch program.

A key aspect of applicant's innovation is automatic generation of executable batch programs from their declarative specifications in the form of models. Our invention views a batch program to comprise of a 'fixed part' and a 'variable part'. The fixed part is common for all batch programs and the variable part is specific to each batch program. Applicant's invention, as claimed, generates the variable part from it's specifications and encapsulates the fixed part in the form of a framework with placeholders where the variable parts can be plugged in. Our framework also

- 5 -

ensures that on failure a batch program will restart automatically, from a consistent state, with minimal loss of computation. The framework also provides automation support for management of resources such as memory.

Applicant points out that Batch fails to teach said code generators and automation for implementing a needed batch program, as claimed. Batch simply teaches a method of executing static batch programs including a plurality of procedure pages consisting of objects and definitions for running and controlling complex auto-startup in a multi-unit production plant while utilizing a uniform control language. Applicant argues that Batch is actually utilizing a control language controlling how the static batch programs execute control over the plant. Batch, either singly or in combination with Wollhaf, fail to teach that data parameters are input into abstractions representing batch functions thereby generating appropriate instances of batch functions and function operations wherein the generated instances are executable as part of a run sequence of the purpose-specific batch program. Applicant respectfully requests the Examiner specifically point out in Batch or Wollhaf where said limitation is taught.

The art of Batch clearly teaches a process control paradigm called 'batch control' that is typically seen in chemical plants, manufacturing plants etc. Applicant's invention has nothing to do with industrial process control. Bristol teaches (pg. 8) The Language models a process hierarchically in terms of its Operations/Objects, modeling the process divisions: the Styrene Plant Operation and the Furnace, Reactor, Heat Recovery, Separator, and Feed Tankage SubOperations. Each Operation is organized into Pages for modeling distinct control functionalities. The example shows several of these Pages H mostly Procedures Pages, describing active control procedures.

In the art of Batch, the framework is the plant operations, therefore, one could

- 6 -

not apply the software of Batch to any other operation without extensive manual programming. Applicant's invention provides a high level mechanism for specifying application-specific variable parts from which their implementations can be automatically generated.

Applicant's claim 1 provides means for generating a batch program through a plurality of abstractions, each representing a batch program; a batch function of the program; operation of the function; a data provider to the function; and an abstraction representing a context class of the function. Applicant's invention also includes a code library which is not found in the art of Batch or Wollhaf as both references fail to teach actually generating and implementing batch programs or code as claimed.

Applicant believes independent claims 1 and 8 are therefore patentable, as amended and argued above. Dependent claims 2-7 and 9-11 are therefore patentable on their own merits, or at least as depended from a patentable claim.

If there are any time extensions required for response in addition to any extension petitioned and paid with this response, such petition is requested, and if there are any fees due over any fees paid with this response, authorization is given to deduct the fees from deposit account 50-0534.

Respectfully submitted,
Vinay Vasant Kulkarni et al.

By /Donald R. Boys/
Donald R. Boys
Reg. No. 35,074

Central Coast Patent Agency, Inc.
3 Hangar Way, Suite D
Watsonville, CA 95076
831-768-1755